Lab 7 Guide

Linear Regression

This lab introduces how to perform linear regression in python with the sklearn library.

For **Question 1**, you can follow the example (Warm-Up: A Model with One Feature). To read the text file in, you can use the following line of code:

```
df = pd.read_csv("AmesHousing.txt", sep='\t')
```

Note that for Q1 we do not need to perform a train test split on the input data, since we are building a regression model from the whole input. So our X_train and y_train can simply be df[['Gr Liv Area']] and df['SalePrice'] respectively. We use double brackets with X_train because the linear regression model's X parameter needs to be in 2 dimensions.

For **Question 2**, you will need to plot the line of predictions using coef_ and intercept_. Here is some sample code to plot a red line from x=500 to x=5000 with coefficient c and intercept b:

```
x = np.linspace(500, 5000, num=5000)
y = [(c*i + b) for i in x]
plt.plot(x, y, c='r')
```

Don't forget to reinclude the scatter plot into the final graph to confirm your line looks correct.

Question 3 is the same as Question 1, but now we have multiple variables in X_train. To make a prediction for a single instance of data, you can initialize a dataframe with a single row and use predict():

```
test = pd.DataFrame({'var1': 1, 'var2': 10, 'var3': 50, 'var4': 100 }, index=[0])
model.predict(test)
```

Question 4 wants to fit another linear regression model. However, all input values need to be **numerical**, so we will need to map any categorical variables to numerical ones (e.g. by using replace()).

Finally for **Question 5**, to specifically ask sklearn to not include an intercept, you can change the fit_intercept parameter:

```
model = LinearRegression(fit_intercept=False)
```